



AGH UNIVERSITY OF KRAKOW

Faculty of Computer Science

X-ray Image Interpretation System

System wspomagający interpretację zdjęć
rentgenowskich

Autorzy: Krzysztof Ćwiertnia, Kacper Kozik, Rafał Piwowar, Michał Sośnik

Kierunek: Informatyka

Promotor: dr. inż. Marcin Kuta

Wprowadzenie

- Współczesna medycyna coraz częściej korzysta z technologii sztucznej inteligencji do wspierania diagnostyki i leczenia.
- Zdjęcia rentgenowskie stanowią jedno z najczęściej używanych narzędzi diagnostycznych w ocenie złamań kości.

Problem:

- Tradycyjna analiza zdjęć rentgenowskich wymaga dużego doświadczenia lekarzy i jest czasochłonna.



Motywacja:

- Zwiększenie dokładności i efektywności diagnostyki rentgenowskiej.
- Ułatwienie pacjentom dostępu do analizy wyników i konsultacji z lekarzami w jednym systemie online.
- Wprowadzenie narzędzi wspierających prywatność danych, takich jak uczenie federacyjne.

Cel pracy i grupy docelowe

Cel:

1. Stworzenie aplikacji webowej, która umożliwia:

- Wykrywanie złamań w wybranych częściach ciała.
- Interakcję użytkownika z modelem: przesyłanie zdjęć, generowanie wyników, analizę obrazu.
- Konsultacje online między pacjentami a lekarzami.

2. Stworzenie dodatkowego modułu dla programistów modeli uczenia maszynowego, umożliwiający dostęp do oznaczonych obrazów bez jakichkolwiek danych osobowych pacjentów, które mogą służyć jako dane do treningu.

Dla kogo przeznaczona jest aplikacja?

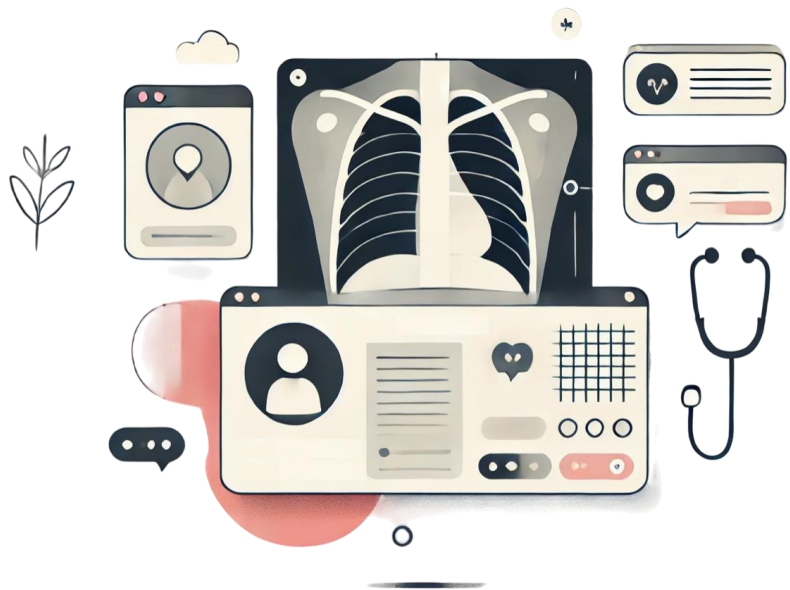


Pacjenci, chcący uzyskać wstępną analizę zdjęć rentgenowskich oraz konsultacje medyczne online.



Lekarze, poszukujący narzędzi do wspomagania diagnostyki i udzielających porad medycznych.

Założenia i zakres projektu



Wymagania funkcjonalne:

- Webowa aplikacja wspomagająca analizę zdjęć rentgenowskich.
- Automatyczna detekcja złamań za pomocą modeli uczenia maszynowego.
- Funkcje: przesyłanie zdjęć, generowanie raportów, konsultacje online.

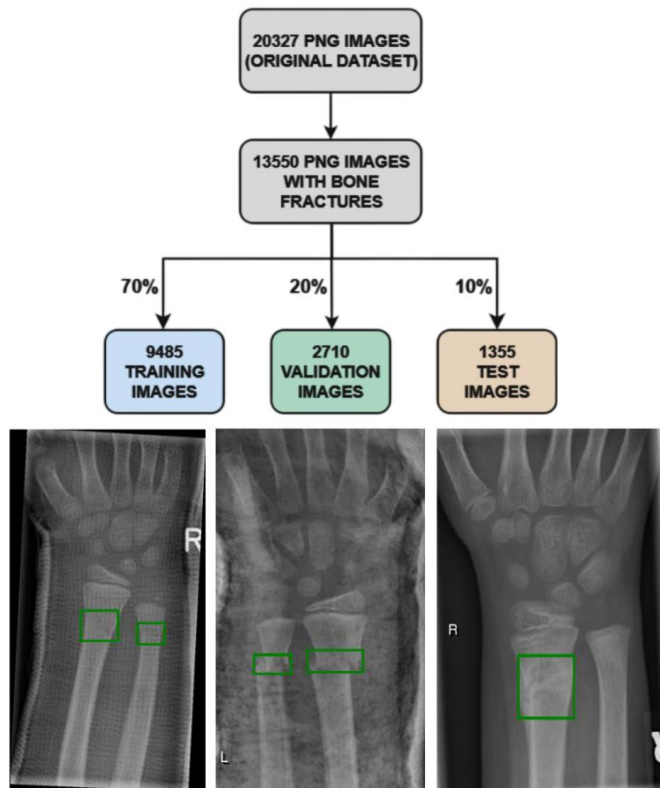
Wymagania niefunkcjonalne :

- Intuicyjność, wydajność, skalowalność, bezpieczeństwo.

Aspekty techniczne i implementacyjne

Zastosowanie głębokich
sieci neuronowych w
detekcji złamań na zdjęciach
rentgenowskich

Zbiór danych zdjęć RTG nadgarstka i przygotowanie do uczenia



- **Zbiór danych ze szpitala uniwersyteckiego w Graz :**
13,550 zdjęć RTG nadgarstka
- Anotacje obejmujące złamania, zapisane jako bounding boxy
- Podział zbioru danych: 70% trening, 20% walidacja, 10% testy
- **Projekcje obrazów:** Dwa różne ujęcia, boczna i posteroanterior (tylno-przednia)

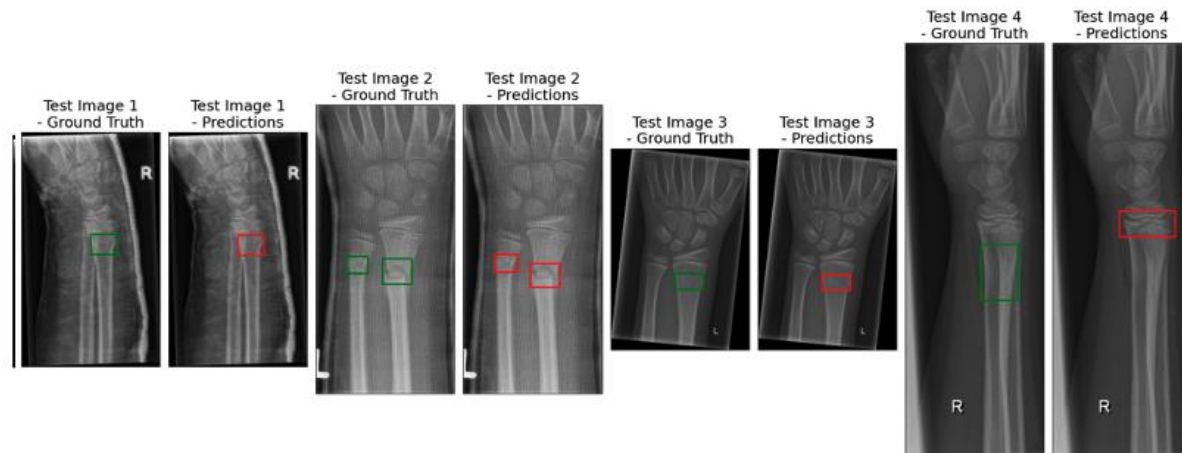
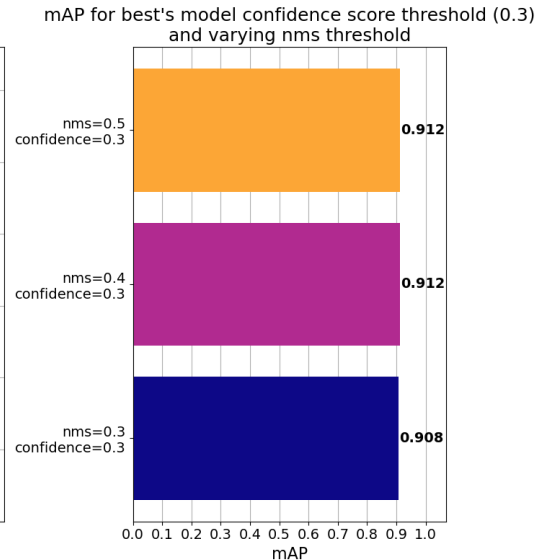
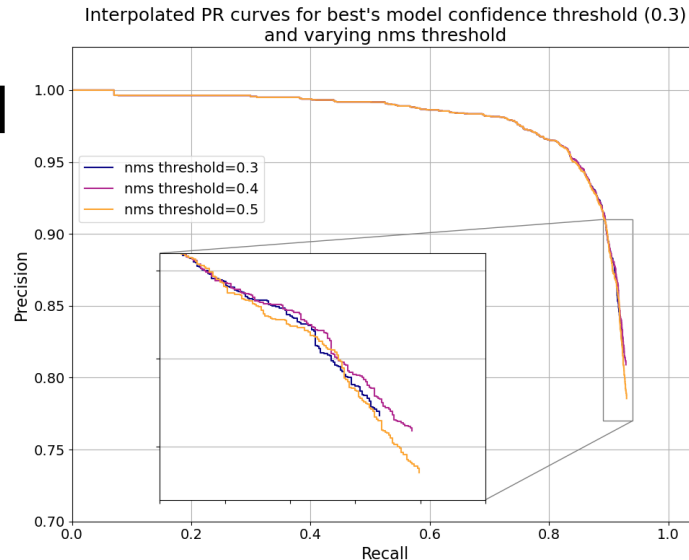
Model Faster R-CNN

Architektura modelu:

- **Backbone:** ResNet-50 (transfer learning ze zbioru COCO).
- **FPN (Feature Pyramid Network):** Połączenie głębokich cech semantycznych z wysoką rozdzielczością.
- **RPN (Region Proposal Network):** Generowanie potencjalnych lokalizacji obiektów (ramki).
- **RoI Pooling:** Normalizacja dla klasyfikacji i regresji ramek.

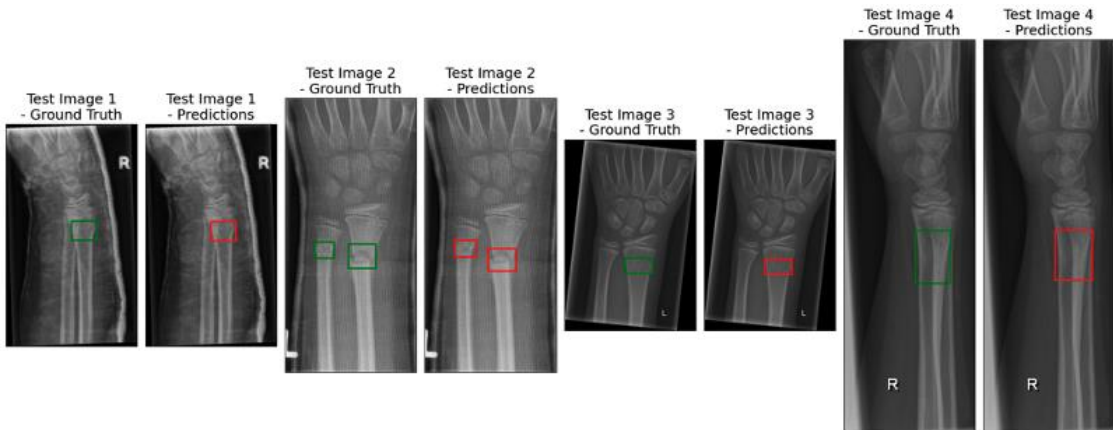
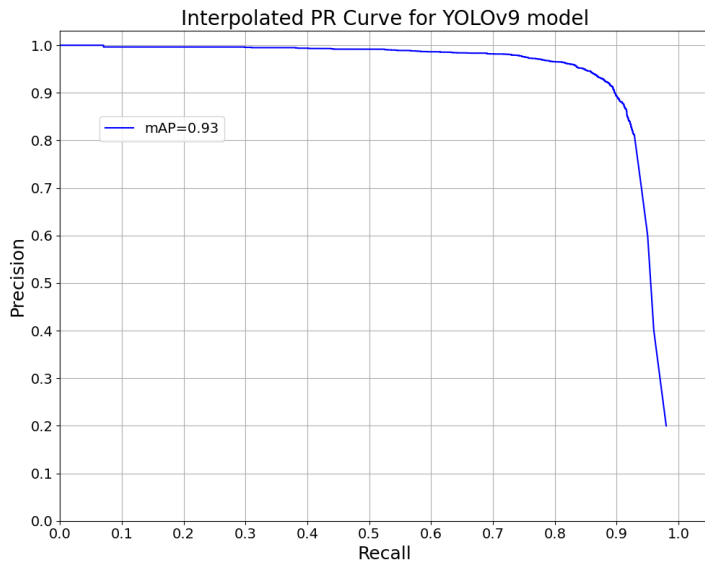
Wyniki:

- **mAP (średnia precyzja):** ok. 0.91
- Krzywe PR + walidacja hiperparametrów (różne progi IoU, Non-Maximum Suppression, Confidence)

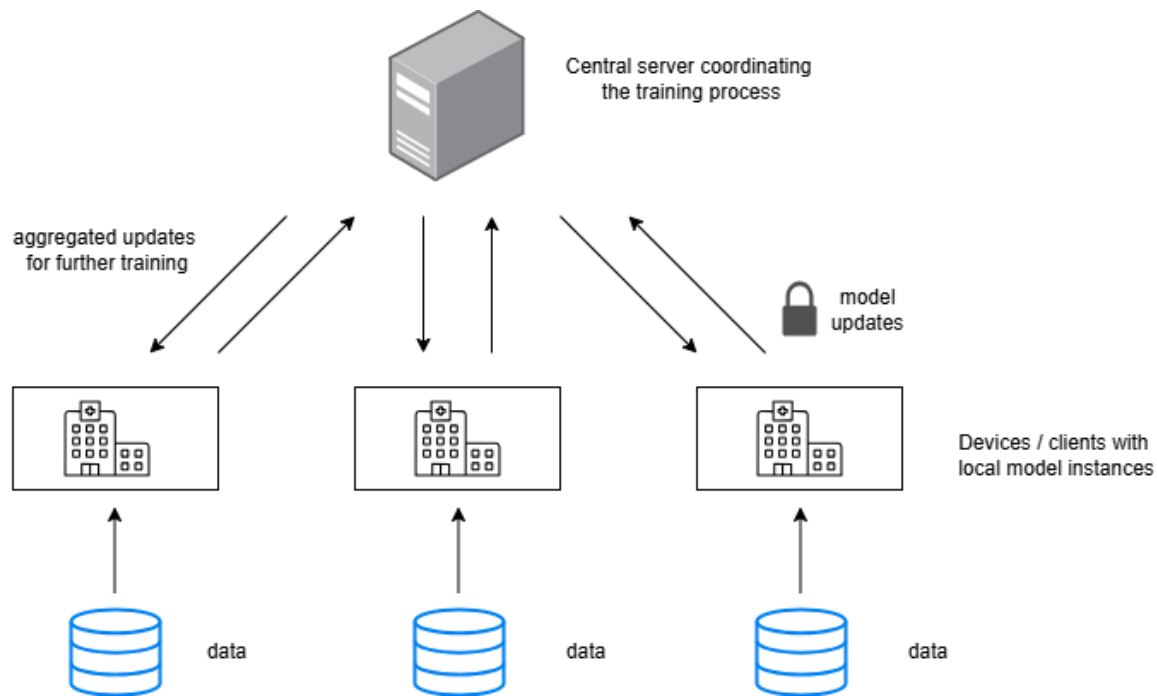


Model YOLO v9

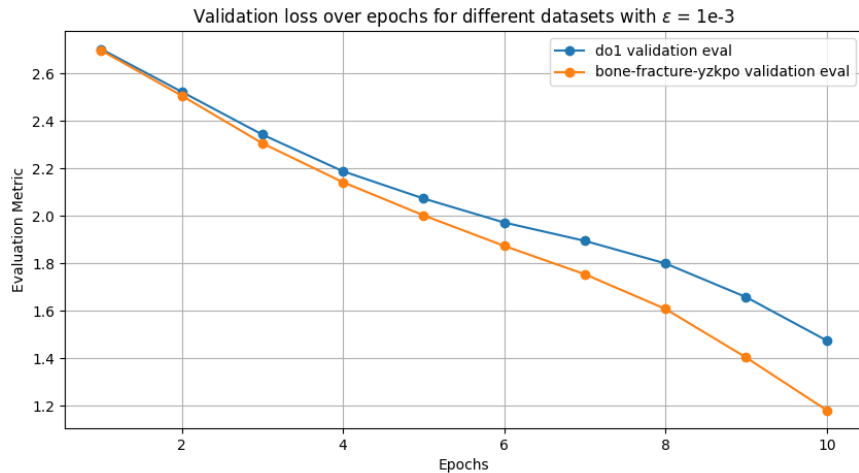
- **Model:** YOLO v9m
- **Trening:** 100 epok, optymalizator SGD
(learning rate = 0.01, batch size = 16)
- **Wyniki:**
 - **mAP (średnia precyzja):** ok. 0.93
 - Krzywa PR bez walidacji hiperparametrów
(duży koszt obliczeniowy)



Uczenie federacyjne

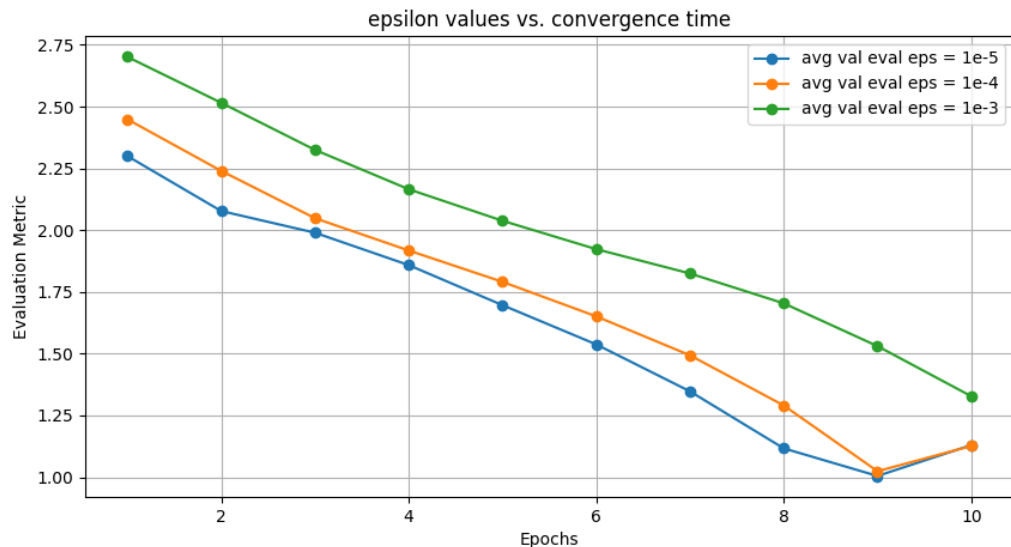


Heterogeniczność danych



- zbiór do1 złamania kości (1019 zdjęć)
- zbiór bone-fractures-yzkpo złamania kości nadgarstka (455 zdjęć)
- W uczeniu federacyjnym dane są z różnych rozkładów
- **Wyniki:** Heterogeniczność danych – mniejszy zbiór danych daje lepsze wyniki

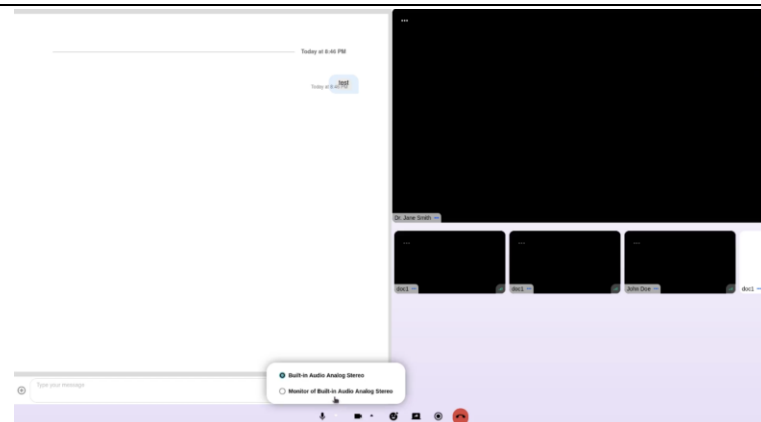
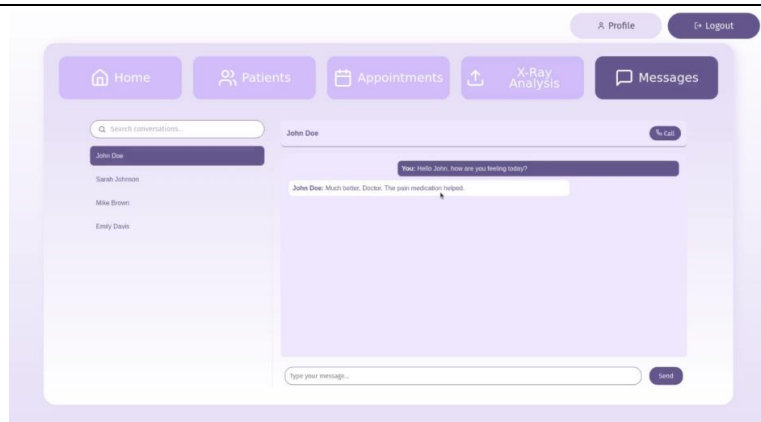
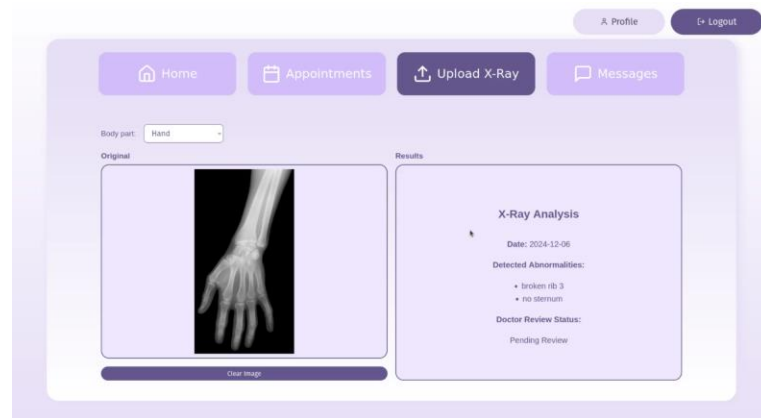
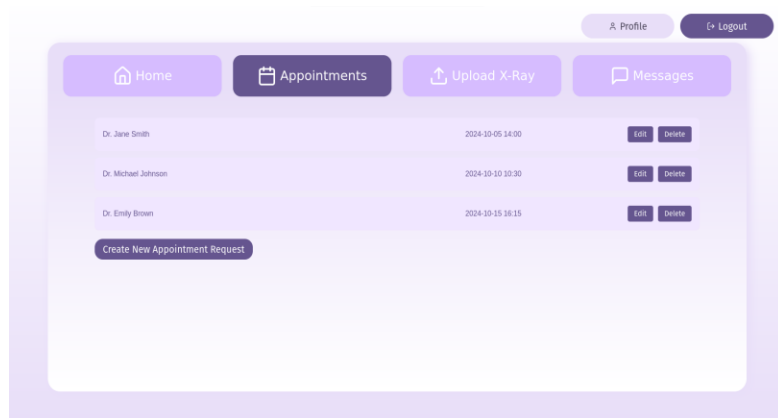
Prywatność różnicowa



- Najczęściej stosowanym rodzajem szumu jest szum Gaussowski.
- Szum jest dodawany w zależności od poziomu ochrony prywatności: wyższy poziom szumu zapewnia silniejszą ochronę prywatności, ale może wpływać na dokładność modelu i czas konwergencji.
- Kompromis między prywatnością a dokładnością kontrolowany jest parametrem ϵ : niższe wartości ϵ oferują lepszą ochronę prywatności.

Frontend

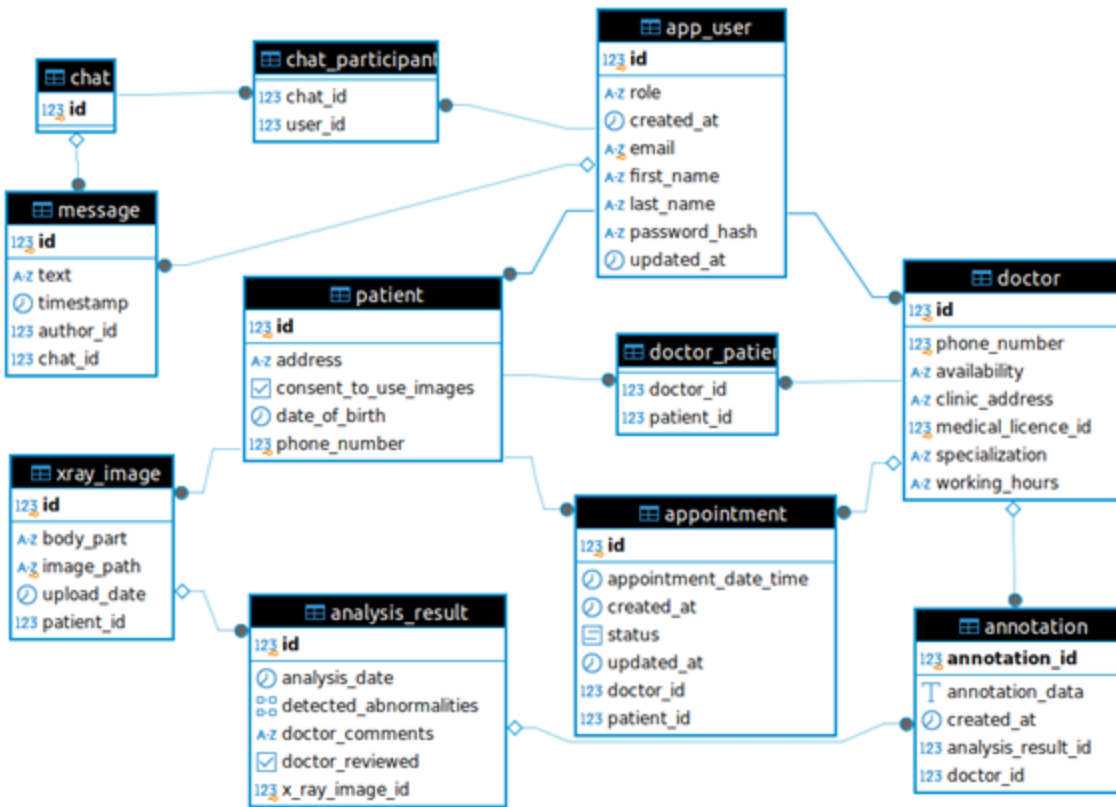
Aplikacja – najważniejsze funkcjonalności



Backend

Baza danych

- Backend Aplikacji korzysta z wbudowanej, plikowej bazy danych H2.
- Do zarządzania bazą używany jest Hibernate — framework ORM. Tabele są tworzone z klas java, jeden obiekt typu entity reprezentuje jeden rekord.
- Relacje między tabelami, są definiowane w polach klas (pacjent posiada listę zdjęć) albo przez dziedziczenie (patient i doctor dziedziczą po user).



API

- Zbudowany w technologii Spring Boot Backend pozwala na szybkie tworzenie REST API i wystawienie go na serwerze.
- API obsługuje operacje **CRUD** na danych w bazie. Zapytania HTTP są interpretowane przez kontrolery, serwisy i repozytoria, trafiając do bazy jako komendy SQL-owe.
- Swagger generuje automatyczną dokumentację API, która jest dostępna w przeglądarce. Dane są przesyłane protokołem HTTP.
- Dodanie **CORS** umożliwia komunikację między aplikacjami działającymi na różnych portach, np. Frontendem, czy modelem AI.

doctor-controller		^
GET	/doctor/{id}	▼
PUT	/doctor/{id}	▼
DELETE	/doctor/{id}	▼
PUT	/doctor/{doctorId}/patients/{patientId}	▼
DELETE	/doctor/{doctorId}/patients/{patientId}	▼
GET	/doctor/	▼
POST	/doctor/	▼
GET	/doctor/by-patient/{patientId}	▼

Zdjęcia w bazie danych

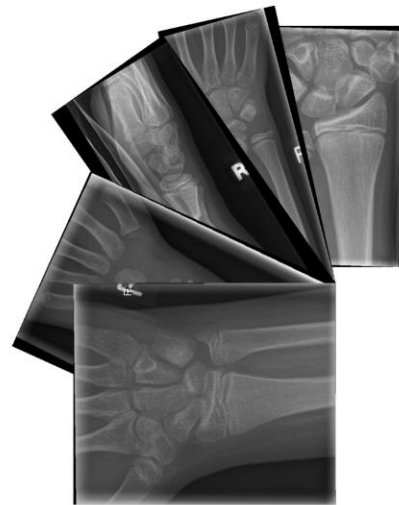
Zdjęcia można traktować jako zasób statyczny, który w trakcie działania aplikacji będzie przesyłany, udostępniany, ale nigdy nie edytowany. Można więc zapisać je na dysku, zaś w bazie przechowywać jedynie URL zdjęcia, razem ze wszystkimi informacjami o nim. Ma to wiele zalet:

- Zdjęcia zajmują dużo przestrzeni. Przechowywanie ich w bazie spowalnia jej działanie,
- Pliki na dysku są łatwo dostępne i nie wymagają specjalnych zapytań SQL,
- Można łatwo zmieniać lokalizację plików bez modyfikacji struktury bazy danych.

Aby poprawnie wyświetlić obraz pod endpointem HTTP, należy dodać odpowiednie nagłówki. W przeciwnym razie dane zostaną błędnie zinterpretowane i wyświetlone jako zwykły tekst.

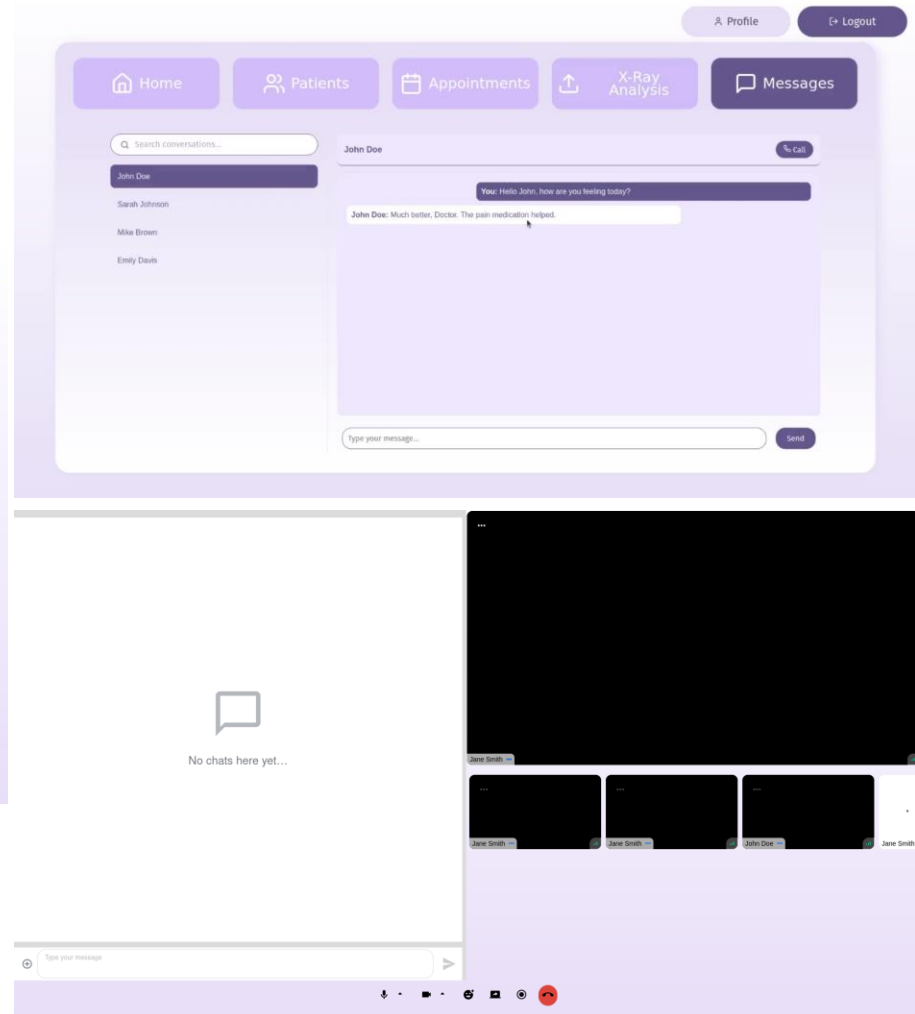
```
headers.add(HttpHeaders.CONTENT_DISPOSITION, "inline; filename=\"" + file.getFilename() + "\"");  
headers.add(HttpHeaders.CONTENT_TYPE, "image/jpeg");  
return new ResponseEntity<>(file, headers, HttpStatus.OK);
```

Samo zdjęcie (file) jest typu Resource.

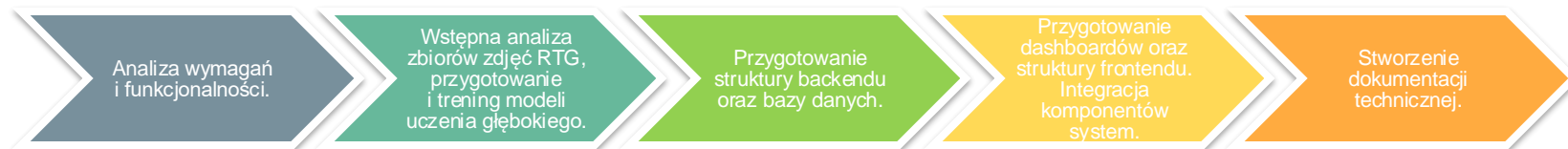


Podsumowanie przebiegu prac

Finalny produkt



Przebieg prac projektowych



Podział prac w zespole

Michał Sośnik:

- Implementacja systemu backendowego
- Obsługa danych medycznych, projekt i implementacja bazy danych
- Wystawienie API oraz kontrola przesyłania zdjęć RTG do analizy automatycznej i użytkowników
- Dodanie komponentów zapewniających bezpieczeństwo aplikacji

Krzysztof Ćwiertnia:

- Stworzenie makiet systemu
- Implementacja frontendu
- Dodanie funkcjonalności czatu oraz wideorozmowy między pacjentem a lekarzem
- Integracja frontendu z backendem

Rafał Piwowar:

- Trening modeli uczenia federacyjnego
- Zastosowaniem Differential Privacy dla ochrony prywatności danych
- Przeprowadzenie eksperymentów z różnymi poziomami ochrony prywatności (parametr ϵ)
- Analiza wpływu parametru ϵ na dokładność i czas konwergencji modelu

Kacper Kozik:

- Planowanie obowiązków w zespole
- Określenie grup docelowych za pomocą User Stories oraz opracowanie wymagań funkcjonalnych
- Zaproponowanie makiet systemu i zdefiniowanie wyglądu oraz funkcjonalności aplikacji
- Analiza i trening modeli Faster R-CNN oraz YOLO v9 do detekcji złamań nadgarstków (zbiór danych GRAZPEDWRI-DX)
- Przeprowadzenie analizy wpływu hiperparametrów modeli na metryki detekcji, takie jak mAP, krzywe PR, precyzja i czułość

Źródła

- Q. Yang, Y. Liu, T. Chen, and Y. Tong. Federated Machine Learning: Concept and Applications. 2019. arXiv: 1902.04885 url: <https://arxiv.org/abs/1902.04885>.

- E. Nagy, M. Janisch, F. HrŹić, and et al. A pediatric wrist trauma X-ray dataset (GRAZPEDWRI-DX) for machine learning. In: Scientific Data 9 (2022), p. 222. doi: 10.1038/s41597-022-01328-z. url: <https://doi.org/10.1038/s41597-022-01328-z>.